ERB-954

CNRC No. 21382

Conseil national de recherches Canada

Division de génie électrique

National Research Council Canada

Division of Electrical Engineering

CARTE D'INTERFACE DMA À QUATRE CANAUX COMPATIBLE AVEC LE BUS S-100-IEEE

P. CHARTIER ET A. DUFOUR

JUIN 1982

Canada

Copyright 1983 par
Conseil national de recherches du Canada

Il est permis de citer de courts extraits et de reproduire des figures ou tableaux du présent rapport, à condition d'en identifier clairement la source.

Copyright 1983 by
National Research Council of Canada

Permission is granted to quote short excerpts and to reproduce figures and tables from this report, provided that the source of such material is fully acknowledged.

Des exemplaires supplémentaires gratuits peuvent être obtenus à l'adresse suivante:

Bureau des publications, Pièce 301
Division de génie électrique
Conseil national de recherches du Canada
Ottawa (Ontario) Canada
K1A 0R8

Additional copies are available free of charge from:

Editorial Office, Room 301
Division of Electrical Engineering
National Research Council of Canada
Ottawa, Ontario, Canada
K1A 0R8

CARTE D'INTERFACE DMA À QUATRE CANAUX COMPATIBLE AVEC LE BUS S-100-IEEE

P. Chartier et A. Dufour

ERB-954
CNRC No. 21382

JUIN 1983

TABLE DES MATIERES

	Page
RÉSUMÉ/ABSTRACT	v
INTRODUCTION	1
LES ELEMENTS DU CIRCUIT	1
Décodage d'adresses	2
Tampons de bus	2
Le contrôleur DMA	2
Arbitrage DMA	3
Signaux de contrôle	3
PROGRAMMATION	3
Transfert	4
Vérification et mise au point	4
RESULTATS PRELIMINAIRES	5
CONCLUSIONS	5
REFERENCES	6
ANNEXE A	7
ANNEXE B	9
ANNEXE C	15
ANNEXE D	19
ANNEXE E	22
ANNEXE F	23

RÉSUMÉ

L'absence sur le marché d'une carte d'interface à accès direct en mémoire pour les micro-ordinateurs utilisant le bus S-100 et la nécessité d'effectuer les transferts à haute vitesse entre un micro-processeur utilisant ce bus et certains montages expérimentaux nous ont amené à développer et à construire la carte DMA4C.

Nous avons développé et mis au point un interface d'usage général permettant les transferts, en lecture ou en écriture, à des vitesses de deux millions d'octets/seconde.

Le présent rapport décrit le circuit et les techniques de montage et représente certains résultats préliminaires démontrant le fonctionnement de l'interface.

ABSTRACT

For microcomputers which use the S-100 bus, there is no direct-memory-access interface card available on the market. Because of this and the need to make high-speed data transfers between a microprocessor using this bus and a number of experimental set-ups, we have developed and fabricated the DMA4C card.

We have designed and developed a general-purpose interface with which it is possible to make data transfers (reading or writing) at speeds of up to two million octets per second.

This report describes the circuit and the assembly techniques used; it also presents a number of preliminary results whch show how the interface operates.

CARTE D'INTERFACE DMA À QUATRE CANAUX COMPATIBLE AVEC LE BUS S-100-IEEE

P. Chartier et A. Dufour

INTRODUCTION

En 1975, la compagnie MITS (maintenant agglomérée à PERTEC Computer Co.) mettait sur le marché le premier microprocesseur pour amateur, le ALTAIR 8800. Cette machine (1) utilisait une structure d'interconnexions (le Bus) à 100 conducteurs appelée S-100. Depuis, nombre de compagnies (on cite le nombre d'environs 50) ont créé des cartes compatibles avec ce bus. Parmi les 100 conducteurs du Bus S-100, certains n'avaient pas été définis, de façon à permettre une croissance du produit, sans devoir changer le bus. Évidemment, maints constructeurs ont dédié ces lignes à des fonctions incompatibles avec le S-100 d'autres constructeurs, amenant un syndrome genre "Tour de Babel" électronique.

Un groupe de travail de IEEE, incluant des manufacturiers de microprocesseurs, présentait en 1979, une proposition pour un standard IEEE du bus S-100 (2). Ils adoptaient eux-même le standard qu'ils proposaient à IEEE et l'incorporaient dans les produits qu'ils offraient à leurs clients.

Le DPS-1 de Ithaca Intersystems est un de ces nouveaux microprocesseurs. Nous l'avons choisi pour servir de mémoire tampon entre une ligne de communication à haute vitesse et un montage électro-optique expérimental.

Grâce au nouveau BUS S-100, le DPS-1 peut dépasser la limite habituelle de 64K octets adressable par le Z-80 qui constitue son unité de calcul, il peut aussi effectuer des transferts par paires de mots, c.-a.-d. 16 bits à la fois au lieu de 8 bits, doublant ainsi le taux de transfert réel. Ces capacités sont exploitées par la carte DMA 4C.

LES ÉLÉMENTS DU CIRCUIT

Le circuit a été conçu à partir d'un contrôleur DMA mono- litique à quatre canaux, le 8237 de la firme INTEL. Cette puce à quarante broches possède deux modes de fonctionnement, soient maître ou esclave. En mode "esclave", le contrôleur reçoit du processeur les paramètres d'opération qui prévaudront au cours des transferts DMA proprement dits. On peut programmer le type de transfert (quatre types) le mode (trois modes, valides pour chaque type) la polarité des signaux de contrôle, le réarmement

automatique en fin de transfert, la stratégie d'arbitrage entre diverses requêtes et quelques autres. Pour chaque canal, on programme le nombre de données ainsi que l'adresse de départ de la zone de mémoire à transférer. Dans ce mode d'opération, le contrôleur se comporte comme un périphérique qu'on peut lire ou auquel on transmet des données.

En mode "maître", le contrôleur prend charge du bus et débranche le processeur. Une fois assuré du contrôle du bus, le contrôleur génère des adresses ainsi que des requêtes en lecture (ou écriture) à la mémoire accompagnées par des écritures (ou lectures) à un des périphériques qu'il commande. A la fin du cycle de transfert, le processeur reprend ses activités au point exact où elles ont été interrompues.

Les divers blocs fonctionnels de DMA 4C permettent d'adapter le 8237 au bus S-100 tel que proposé à l'IEEE. On retrouve donc les fonctions de:

(a) décodage d'adresses
(b) tampons de Bus
(c) contrôleur DMA
(d) arbitrage DMA
(e) signaux de contrôle

Décodage d'adresses

Les circuits (21A, 24, 15A) ont pour fonction d'activer le 8237 en mode "esclave". Dans ce mode, la carte se comporte comme un port d'entrée-sortie. Le nombre de fonctions et d'options du 8237 est tel qu'il doit couvrir une plage d'adresses (16 en tout) pour recevoir toutes les commandes. Pour ce faire, il décode lui-même les bits d'adresses A0 à A3 les bits A4 à A7 sont démultipliés et une des sorties du décodeur est choisie à l'aide d'un cavalier pour activer le signal CS du contrôleur.

Tampons de bus

Les règles de l'art exigent qu'on isole du reste du système les composantes d'un bloc fonctionnel. Les puces (5, 6, 1, 2, 3, 4 et 20) permettent au contrôleur d'accéder au bus pour lire ou écrire tout en gardant la possibilité de s'en détacher lorsque son intervention n'est pas requise.

Le contrôleur DMA

Le circuit 8237 de Intel est un contrôleur d'accès direct à la mémoire à haute performance. Sa fonction est de générer, lors d'un transfert DMA, une séquence d'adresses et de signaux de contrôle de façon à établir la communication entre des blocs de mémoire et des régistres de

lecture ou d'écriture. Lorsqu'un transfert est requis, le contrôleur envoie au processeur (le Z80) un signal pHOLD. Dès qu'il a terminé le traitement de l'instruction courante, le processeur envoie un récépisse pHLDA et tombe dans un état d'activité suspendue. À partir de cet instant, le contrôleur a tous les attributs du processeur dont il joue le rôle. Il génère donc des adresses, des signaux de lecture et d'écriture. Quand un nombre approprié de transferts ont été effectués, le contrôleur tombe en "hibernation" jusqu'au transfert suivant.

Le 8237 ne possédant que huit (8) broches d'adresses doit émettre ces adresses en deux étapes les huit BPS (8 bits plus significatifs) sont mis en mémoire dans un régistre 8282 avant d'être lancés sur le bus d'adresses. Ce régistre est mis à jour chaque fois qu'une nouvelle page de 256 bits est entamée. Le 8237 coordonne le fonctionnement du 8282 à l'aide de la broche AEN (Address Enable) et ADSTB (Address Strobe).

Arbitrage DMA

Le bus S-100 permet la cohabitation de 16 contrôleurs DMA. Lorsqu'un système compte plus qu'un seul contrôleur, il est nécessaire d'établir un protocole régissant l'accès au bus par ordre de priorité des contrôleurs. Ce protocole est cablé dans le bloc fonctionnel d'arbitrage DMA. La proposition IEEE S-100 (2) décrit avec précision le mode d'arbitrage DMA et présente un circuit le réalisant. Nous avons retenu une version modifiée du circuit suggéré, ayant strictement le même fonctionnement. Cette fonction est réalisée à l'aide des puces (3, 11, 12, 13, 21, 14, 17 et 15).

Signaux de contrôle

Le bus S-100 reconnaît un certain nombre de signaux servant à contrôler et à coordonner l'action des divers éléments du système. Ce sont: pHOLD* pSYNC, pWR*, pDBIN, CDSB*, SDSB*, sWO, sM1, DODSB*, sINP, sOUT; le contrôleur DMA 4C doit générer ces signaux à partir, essentiellement, des cinq suivants: HLDA, IOW*, IOR*, MEMW*, MEMR*. Les puces (10, 7, 8, 15, 16, 25) permettent au 8237 de donner aux périphériques les signaux adéquats prévus par le standard S-100.

PROGRAMMATION

Le contrôleur, pour effectuer des transferts, doit être programmé convenablement. Cette programmation s'échelonne sur six étapes distinctes.

La première consiste en une réinitialisation globale, le "Master Clear". On réarme du même coup une bascule qui détermine l'ordre d'entrée

des données. En effet, les adresses et la dimension des blocs à transférer sont envoyés sous forme de double mots (16 bits) et leur ordre d'arrivée détermine leur valeur.

La deuxième étape est la programmation, en deux transferts, de l'adresse de départ. La troisième encore en deux étapes initialise le compteur de mots à la valeur de la longueur du bloc à transférer.

Durant la quatrième étape de programmation, on établit le contenu du régistre de commande. Le contenu de celui-ci définit le type de transfert à effectuer et la stratégie régissant la priorité des requêtes simultanées. Ces informations affectent le fonctionnement global du contrôleur.

L'avant dernière étape, la cinquième, établit pour les canaux impliqués dans un transfert le mode de transfert qui prévaudra. Les informations qu'on place dans ce régistre n'affectent que le canal qui y est associé.

On a accès, sur le 8237 à quatre canaux de communication. Le contrôleur peut activer un de ces canaux lorsqu'il reçoit une requête en ce sens. La requête peut provenir de l'extérieur par les lignes DRQØ, DRQ1, DRQ2, ou DRQ3, soit du processeur, par programmation du régistre des requêtes. On peut activer individuellement certains canaux pour permettre des transferts. Lors de la réinitialisation, tous les canaux sont désactivés. En programmant des zéros dans les bits appropriés du régistre masque ou réactive les canaux. C'est la sixième et dernière étape du processus d'initialisation.

Transfert

Dans l'exemple de la figure C2, on peut voir une septième étape dans laquelle le processeur génère une commande par logiciel. À partir de cet instant, le contrôleur peut et va prendre charge du bus, transférer des données et rendre le bus, avant que le processeur n'exécute l'étape suivante qui est une boucle d'attente sans fin.

Vérification et mise au point

Pour la mise au point et la vérification du contrôleur, nous avons ajouté sur la même carte que celui-ci, un régistre de sortie branché sur les signaux de commande du canal Ø. L'ensemble a été monté sur une carte universelle format S-100 et cablé à l'aide du système U-contact (TM) de la firme 3M. Pour réduire le bruit électrique l'alimentation des diverses puces de la carte était faite à l'aide des conducteurs type "Bus-bars" de Rodgers Corp.

Les essais préliminaires ont été réalisés en installant le contrôleur dans un microprocesseur IMSAI 8800 dans sa configuration originale. Le microprocesseur, un 8080 était remplacé par un ICE-80 branché au système de développement de la section de Génie électronique, un MDS-800.

La programmation a été faite en PLM, langage évolué semblable au PL/1 et fonctionnant sur MDS-800. Le module ICE-80 permet de suivre le comportement du processeur d'un système; lors d'un transfert, ce même processeur cesse toute activité. Nous avons eu recours à un analyseur logique LAM-3250 de DOLCH instruments pour observer le fonctionnement du contrôleur lui-même au moment des transferts.

RÉSULTATS PRÉLIMINAIRES

En utilisant le 8Ø8Ø, dont l'horloge fonctionne à 2 MHz, on a pu effectuer des transferts à 1 Mega-octet par seconde, en mode comprimé tel que montré par la figure 2. Tous les signaux de contrôle fonctionnent selon leurs spécifications et le niveau de bruit, sur le bus, reste dans les limites acceptables.

La charge électrique de la carte est de 680 mA sur la ligne de 8 V non régularisée.

Il est à prévoir que ces résultats vont changer lors des essais à bord du DPS-1 dont l'horloge fonctionne à 4 MHz et que tous les régistres de sortie et d'entrée seront branchés.

CONCLUSIONS

Le circuit est très général et parfaitement adapté aux besoins que nous avions. La relative nouveauté du BUS S-100 IEEE a causé des incertitudes dans l'interprétation de certaines lignes. Il est évident que le circuit gagnerait en versatilité si on y ajoutait une unité de traduction d'adresses, de façon à couvrir tout l'espace adresse du nouveau BUS S-100, mais nous n'avions pas besoin de cette commodité dans notre application.

Il est regrettable de plus, que la structure du 8237 rende incompatibles, sur une même carte les modes de transfert mémoire à mémoire et mémoire à périphérique. En effet, les bus de données S-100 sont unidirectionnels et dans un transfert de mémoire à périphérique, la carte de contrôle n'a aucune interaction avec eux et on se contente de monter le contrôleur comme un périphérique afin de le programmer. Lors du transfert

de mémoire à mémoire, la carte se comporte comme un nouveau CPU, ce qui exige une connexion différente des Bus de Données.

Finalement, le fait que les adresses générées par le 8237 soient multiplexées exige qu'à tous les 256 transferts, on introduise un cycle de plus afin de changer les bits les plus significatifs. Ceci empêche un flot continu de données à vitesse fixe. Ce problème inhérent à l'architecture du 8237 ne possède aucune solution.

REFERENCES

1. Bursky, D. The S-100 Bus HANDBOOK. Hayden Book Company, Inc., Rochelle Park, NY. 1980.

2. Standard Specification for S-100 Bus interface devices, COMPUTER, July 1979, IEEE TASK 696.1/D2.

3. Pour les chips 8237, 8282, Component data catalog, Intel Corp., Literature Department, Santa Clara, CA. 1980.

ANNEXE A: LISTE DES COMPOSANTES

(1C#)	(#Pins)	(I.C.I.D.)	# of gates used	Name
1.	16-	74LS367	6/6	4 lines & 2 line enable inputs (true data)
2.	16-	74LS368	5/6	4 lines & 2 line enable inputs (inv. data)
3.	14-	74LS03	3/4	Quadruple 2 inputs nand gates
4.	16-	74LS368	3/6	4 lines 2 lines enable inputs (inv. data)
5.	16-	8216	1/1	4 bits parallel bi-directional bus
6.	16-	8216	1/1	4 bits parallel bi-directional driver
7.	14-	74LS00	3/4	Quadruple 2 input nand gates
8.	16-	74LS367	2/6	4 lines & 2 lines enable inputs (true data)
9.	14-	74LS32	1/3	Quadruple 2 input or gates
10.	14-	74LS00	4/4	Quadruple 2 input nand gates
11.	14-	74LS08	4/4	Quadruple 2 input and gates
12.	14-	74LS32	4/4	Quadruple 2 input or gates
13.	14-	74LS05	6/6	Hex inverters O.C.
14.	14-	74LS74	2/2	Dual D type pos edge trig F/F
15.	14-	74LS27	3/3	Triple 3 input nor gates
16.	14-	74LS74	2/2	Dual D type pos edge trig F/F
17.	14-	74LS20	2/2	Dual 4 input nand gates
19.	16-	Load res.		Resistor pack
20.	20	8282		Octal latch
21.	14-	74LS04	6/6	Hex inverters
22.	14-	74LS04	6/6	Hex inverters
23.	40	8237-2	1/1	DMA controller
24.	24	74154	1/1	4 line to 16 line decoders/Demult
26.	16-	On off sn		

Numéro des pièces	MAGASIN N-50	QUANTITÉ
1 - 74LS367	# 586-35-350-3845	2
2 - 74LS368	# 586-35-350-3855	2
3 - 7403	# 586-20-200-2280	1
4 - 8216	COMMANDE SÉPARÉE	2
5 - 74LS00	# 586-35-350-3510	2
6 - 74LS08	# 586-35-350-3540	1
7 - 74LS32	# 586-35-350-3600	1
8 - 7405	# 586-20-200-2300	1
9 - 74LS74	# 586-35-350-3640	2
10 - 74LS27	# 586-35-350-3585	2
11 - 74LS20	# 586-35-350-3582	1
12 - Res Pack	COMMANDE SÉPARÉE	1
13 - 8282	COMMANDE SÉPARÉE	1
14 - 74LS04	# 586-35-250-3530	2
15 - 8237-2	# 586-20-200-2750	1
16 - 74154	# 586-35-350-3583	1
17 - 74LS21		1

ANNEXE B: MEMDM3, PROGRAMME DE MISE AU POINT

```
PL/I-80 V1.3  COMPILATION OF  MEMDM3

B: Built-In Subroutine Trace
I: Interlist Source and Code
N: Nesting Level Display
S: Symbol Table List

   1 a      MEMDM3
   2 a
   3 a
   4 a
   5 b            PROCEDURE OPTIONS  MAIN ;
   6 b
   7 b         DECLARE
   8 b             OUTP ENTRY  BIT 8 ,BIT 8   EXTERNAL;
  10 b
  11 b      *LIRE KEYIES DE LA CAMERA ET EMMAGASINER EN MEMOIRE A L'ADRESSE
              SPECIFIE PAR CURR ADR2 *
                  12 b
                  13 b
  14 b            %REPLACE MST_CLR BY '80'B4;
  15 b            %REPLACE T_MST_CLR BY '8D'B4;
  16 b
  17 b
  18 b            %REPLACE T_CLR_FL_FF BY  8C'B4;
  19 b
  20 b
  21 b            DECLARE CURAD POINTER
  22 b            CURR_ADR2 BIT  16  BASED  CURAD ,
  23 b            CURR_REG2  2  BIT  8  BASED  CURAD ;
  24 b            %REPLACE T_CURR_ADR2 BY '84'B4;
  25 b
  26 b
  27 b
  28 b            DECLARE CUWRD POINTER.
  29 b            CURR_WRD2 BIT  16  BASED  CUWRD .
  30 b            CURR_WREG2  2  BIT  8  BASED  CUWRD ;
  31 b            %REPLACE T_CURR_WRD2 BY '85'B4;
  32 b
  33 b
  34 b            DECLARE MODE_REG BIT  8 ;
  35 b            %REPLACE T_MODE_REG BY '8F'B4;
  36 b
  37 b
```

```
38 b
39 b          DECLARE MSK_REG BIT  8 ;
40 b          %REPLACE T_MSK_REG BY '8'B4;
41 b
42 b
43 b
44 b          DECLARE CMD_REG BIT  8 ;
45 b          %REPLACE T_CMD_REG BY '88'B4;
46 b
47 b
48 b          DECLARE REQ_REG BIT  8 ;
49 b          %REPLACE T_REQ_REG BY '93'B4;
50 b
51 b               *SEQUENCE D EXECUTION DU PROGRAMME*/
52 b
53 b
54 b               CALL OUTP T_MST_CLR.MST_CLR ;
55 b
56 b               CALL OUTP T_CLR_FL_FF.MST_CLR ;
57 b
58 b          CURR_ADR2='8000'B4;
59 b               CALL OUTP T_CURR_ADR2.CURR_REG2 1  ;
60 b               CALL OUTP T_CURR_ADR2.CURR_REG2 2  ;
61 b
62 b          CURR_ARD2 '2000'B4;
63 b               CALL OUTP T_CURR_ARD2.CURR_*REG2 1  ;
64 b               CALL OUTP T_CURR_ARD2.CURR_*REG2(2) ;
65 b
66 b          CMD_REG  '00'B4;
67 b               CALL OUTP T_CMD_REG.CMD_REG ;
68 b
69 b          MODE_REG '66'B4;
70 b               CALL OUTP T_MODE_REG.MODE_REG ;
71 b
72 b          MSK_REG 'FE'B4;
73 b               CALL OUTP T_MSK_REG.MSK_REG ;
74 b
75 b          REQ_REG  '0'B4;
76 b               CALL OUTP T_REQ_REG.REQ_REG ;
77 b
78 b
79 b     LOOP    GO TO LOOP;
80 b
81 b     END;
```

```
   SYMBOLS

BLOCK AT LINE 1  AUTO STORAGE 0 BYTES
a 0000 MEMDM3 ENTRY PARAMETERS 0  EXTERNAL CONSTANT
BLOCK AT LINE 7  AUTO STORAGE 6 BYTES
c 0000 OUTP ENTRY PARAMETERS 2  EXTERNAL CONSTANT
c 0000 .* BIT(8)  PARAMETER
c 0001 .* BIT(8)  PARAMETER
c 0002 CURAD POINTER AUTOMATIC
c 0000 CURR_ADR2 BIT(16)  BASED CURAD
c 0000 CURR_REG2(1:2)  BIT(8)  BASED CURAD
c 0002 CUWRD POINTER AUTOMATIC
c 0004 CURR_WRD2 BIT(16)  BASED CUWRD
c 0000 CURR_WREG2(1:2)  BIT(8)  BASED CUWRD
c 0004 MODE_REG BIT(8)  AUTOMATIC
c 0005 MSK_REG BIT(8)  AUTOMATIC
c 0006 CMD_REG BIT(8)  AUTOMATIC
c 0007 REC_REG BIT(8)  AUTOMATIC
c 0002 LOOP LABEL CONSTANT
   NO ERROR(S) IN PASS 1

   NO ERROR(S) IN PASS 2



PL/I-80 V1.3  COMPILATION OF: MEMDM3

   1 a 0000 MEMDM3
       0000         LXI   H,0000
       0003         CALL  ?START
   2 a 0006
   3 a 0006
   4 a 0006
   5 a 0006           PROCEDURE OPTIONS  MAIN ;
   6 a 0006
   7 c 0006       DECLARE
   8 c 0006           OUTP ENTRY  BIT(8)  BIT(8)   EXTERNAL;
   9 c 0006
  10 c 0006
  11 c 0006    *LIRE KBYTES DE LA CAMERA ET EMMAGASINER EN MEMOIRE A L'ADRESSE
                 SPECIFIE PAR CURR_ADR2 *
  12 c 0006
  13 c 0006
  14 c 0006       %REPLACE MST_CLR BY  00'B4;
  15 c 0006       %REPLACE T_MST_CLR BY '0D'B4;
  16 c 0006
  17 c 0006
  18 c 0006       %REPLACE T_CLR_FL_FF BY '8C'B4;
  19 c 0006
  20 c 0006
```

```
 21 c 0000      DECLARE CURAD POINTER.
 22 c 0000      CURR ADR2 BIT  16  BASED  CURAD ,
 23 c 0000      CURR REG2  2  BIT  8  BASED  CURAD ;
 24 c 0000      %REPLACE I CURR ADR2 BY '84'B4;
 25 c 0000
 26 c 0000
 27 c 0000
 28 c 0000      DECLARE CUWRD POINTER
 29 c 0000      CURR WRD2 BIT  16  BASED  CUWRD ,
 30 c 0000      CURR WREG2  8  BIT  8  BASED  CUWRD ;
 31 c 0000      %REPLACE I_CURR WRD2 BY '85'B4;
 32 c 0000
 33 c 0000
 34 c 0000      DECLARE MODE REG BIT  8 ;
 35 c 0000      %REPLACE I MODE REG BY '86'B4;
 36 c 0000
 37 c 0000
 38 c 0000
 39 c 0000      DECLARE MSK REG BIT  8 ;
 40 c 0000      %REPLACE I_MSK_REG BY '87'B4;
 41 c 0000
 42 c 0000
 43 c 0000
 44 c 0000      DECLARE CMD REG BIT  8 ;
 45 c 0000      %REPLACE I CMD REG BY '88'B4;
 46 c 0000
 47 c 0000
 48 c 0000      DECLARE REC_REG BIT  8 ;
 49 c 0000      %REPLACE I REC REG BY '89'B4;
 50 c 0000
 51 c 0000           *SEQUENCE D'EXECUTION DU PROGRAMME*
 52 c 0000
 53 c 0000
 54 c 0000        CALL OUTP I MSI CLR MST CLR ;
      0000        LXI  H,* A 0011
      0003        CALL OUTP
 55 c 0006
 56 c 0006        CALL OUTP I CLR EI FF MST CLR ;
      0006        LXI  H,* A 0015
      0009        CALL OUTP
 57 c 0012
 58 c 0012      CURR_ADR2='0002'B4;
      0012        LHLD CURAD A 0029
      0015        MVI  M 02
      0017        INX  H
      0018        MVI  M 00
 59 c 001A        CALL OUTP I CURR ADR2 CURR_REG2 1  ;
      001A        LHLD CURAD A 0029
      001D        SHLD * A 001F
      0020        LXI  H,* A 0019
```

```
      0023          CALL OUTP
60 c  0026            CALL OUTP T CURR ADR2.CURR REG2 2  ;
      0026          LHLD CURAD + 0009
      0029          INX  H
      002A          SHLD * + 001F
      002D          LXI  H,* + 001D
      0030          CALL OUTP
61 c  0033
62 c  0033        CURR_WRD2 = '2000'B4;
      0033          LHLD CURRD + 0005
      0036          MVI  M,20
      0038          INX  H
      0039          MVI  M,20
63 c  003B            CALL OUTP T CURR_WRD2.CURR_WREG2(1) ;
      003B          LHLD CURRD + 000B
      003E          SHLD * + 0023
      0041          LXI  H,* + 0021
      0044          CALL OUTP
64 c  0047            CALL OUTP T CURR WRD2.CURR WREG2 2  ;
      0047          LHLD CURRD + 000B
      004A          INX  H
      004B          SHLD * + 0027
      004E          LXI  H,* + 0025
      0051          CALL OUTP
65 c  0054
66 c  0054        CMD_REG = '00'B4;
      0054          LXI  H,CMD_REG + 000B
      0057          MVI  M,00
67 c  0059            CALL OUTP T CMD_REG CMD_REG ;
      0059          LXI  H,* + 0029
      005C          CALL OUTP
68 c  005F
69 c  005F        MODE_REG = '86'B4;
      005F          LXI  H,MODE_REG + 000D
      0062          MVI  M,86
70 c  0064            CALL OUTP T MODE_REG MODE_REG ;
      0064          LXI  H,* + 002D
      0067          CALL OUTP
71 c  006A
72 c  006A        MSK_REG = '0F'B4;
      006A          LXI  H,MSK_REG + 0003
      006D          MVI  M,0F
73 c  006F            CALL OUTP T MSK_REG.MSK_REG ;
      006F          LXI  H,* + 0031
      0072          CALL OUTP
74 c  0075
75 c  0075        REQ_REG = '06'B4;
      0075          LXI  H,REQ_REG + 0010
      0078          MVI  M,06
76 c  007A            CALL OUTP T REQ_REG.REQ_REG ;
```

```
        007A            LXI  H,* A 0035
        007D            CALL CUIF
    77 c 0080
    78 c 0080
    79 c 0083 LOOP      GO TO LOOP;
        0086            JMP  LOOP
    80 c 0083
    81 a 0083 END;

CODE SIZE    0083
DATA AREA    0039
FREE SYMS    36C1
END  COMPILATION
```

CP/M 2.2 Ithaca InterSystems Inc. BIOS version 4.0 (standard 3 11 81

A

ANNEXE C: SIGNAUX BINAIRES EN MODE TRANSFERT

Canal	1	A3	35
	2	A2	34
	3	A1	33
	4	A0	32
	5	HRQ	10
	6	HLDA	7
	7	$\overline{\text{MEMR}}$	3
	8	$\overline{\text{ADSB}}$	1C 22 pin 4
	9	$\overline{\text{DDSB}}$	1C 22 pin 2
	10	$\overline{\text{SDSB}}$	1C 22 pin 6

MEMDM 3. LOC
7- $\overline{\text{MEMRN}}$ 4

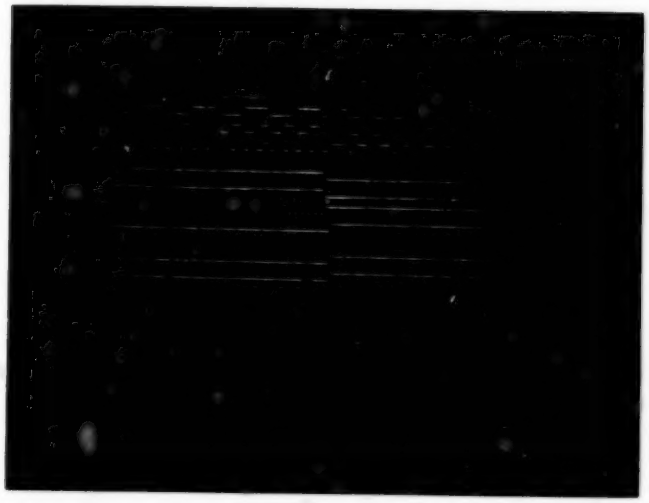


Figure C1.

1	A3	35	
2	A2	34	
3	A1	33	
4	A0	32	
5	HRQ	10	
6	HLDA	7	
7	$\overline{\text{DACK 2}}$	14	
8	READY	6	
9	$\overline{\text{IOR}}$	1	
10	$\overline{\text{MEMW}}$	4	
11	1c14	Pin 9	cont. READY
12	1c14	Pin 14	CLEAR
13	1c14	Pin 1	EOF
14	1c14	Pin 13	STROBE
15			
16			

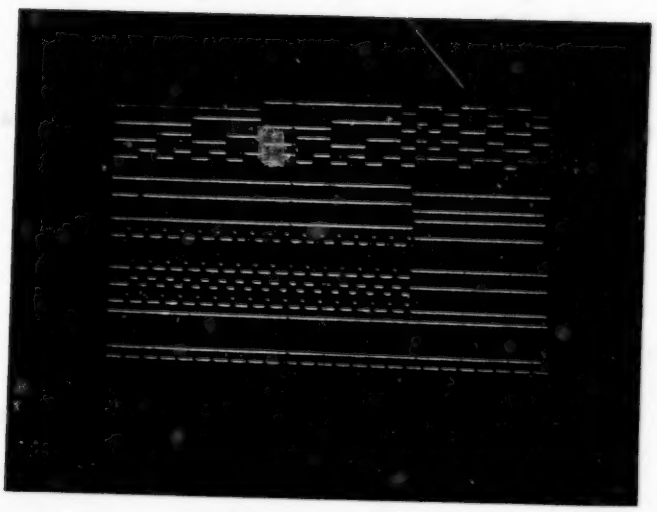


Figure C2.

		8237	
1	A3	35	
2	A2	34	
3	A1	33	
4	A0	32	
5	HRQ	10	
6	HLDA	7	
7	$\overline{\text{DACK 2}}$	14	
8	READY	6	
9	$\overline{\text{IOR}}$	1	
10	$\overline{\text{MEMW}}$	4	
11	1c14	Pin 9	cont. READY
12	1c14	Pin 14	CLEAR
13	1c14	Pin 1	EOF
14	1c14	Pin 13	STROBE
15			
16			

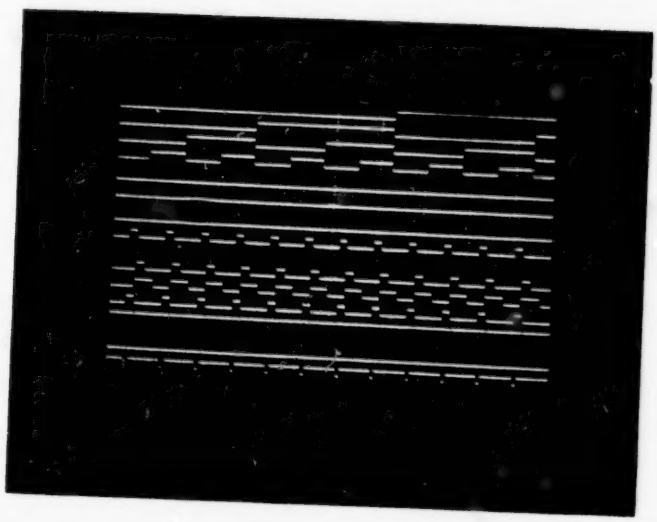
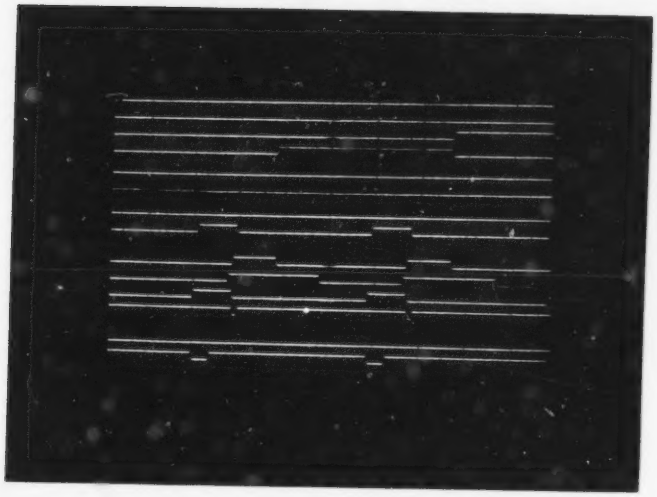
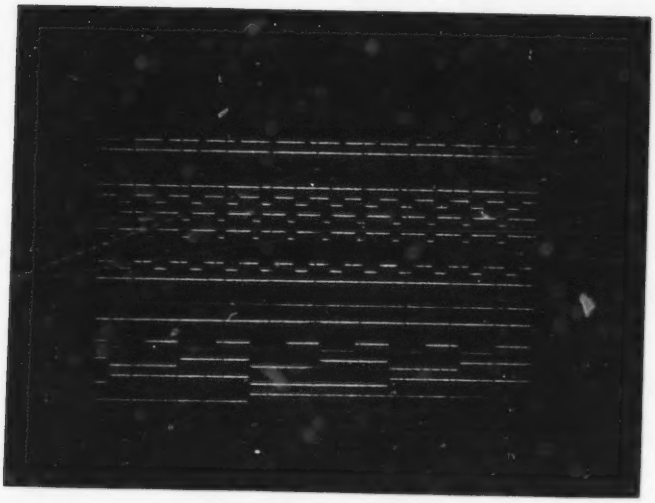


Figure C3.

Mêmes signaux que la 50 Ns (horlage)



Même séquence que 4 mais image extended (×5)

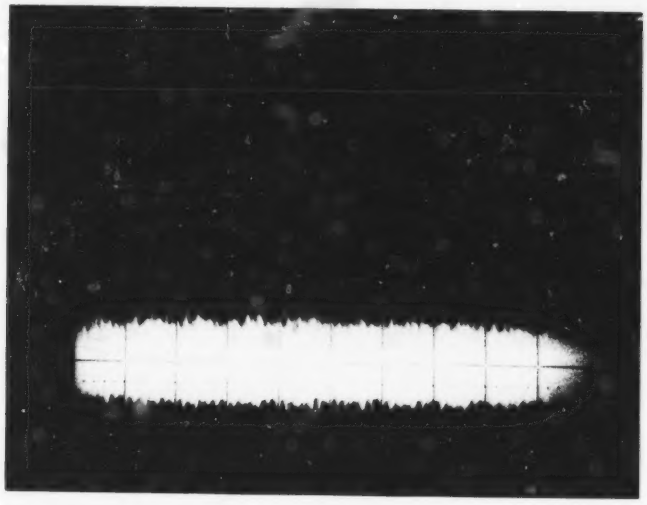


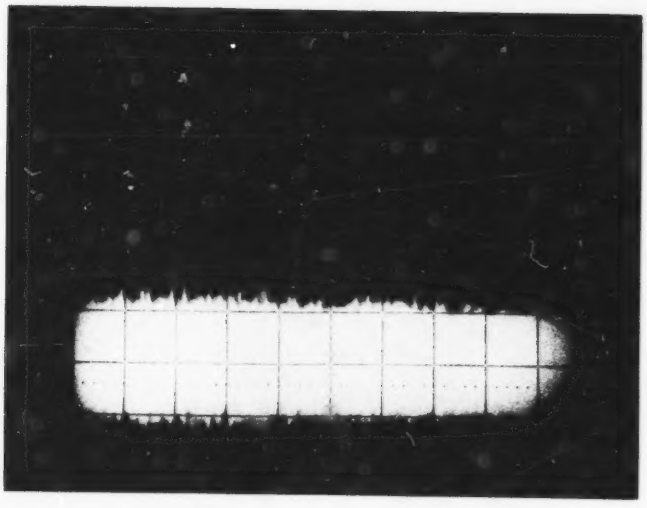
Même séquence que A mais avec le bit 3 du Command Register changé à /1 (compressed timing)

ANNEXE D: NIVEAUX DE BRUIT ET QUALITÉ DU SIGNAL

50 mV/div = 100 mV/bruit
time base

Prise aux bornes du régulateur avec la carte DMA inactive

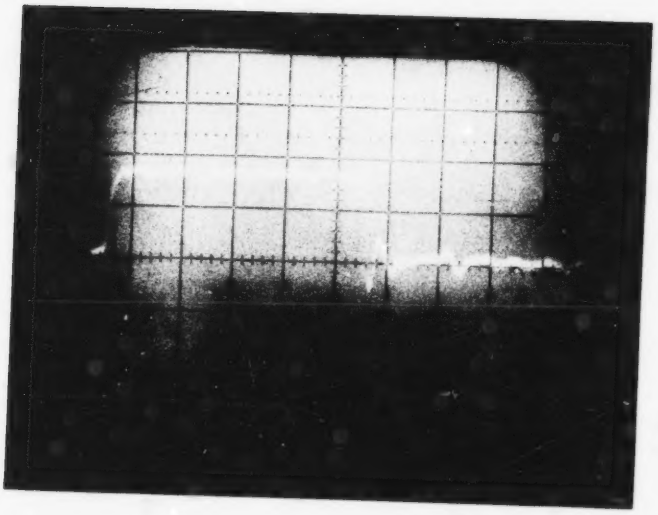




200 mV de bruit en condition dynamique.
(Séquence de programmataion du 8237)

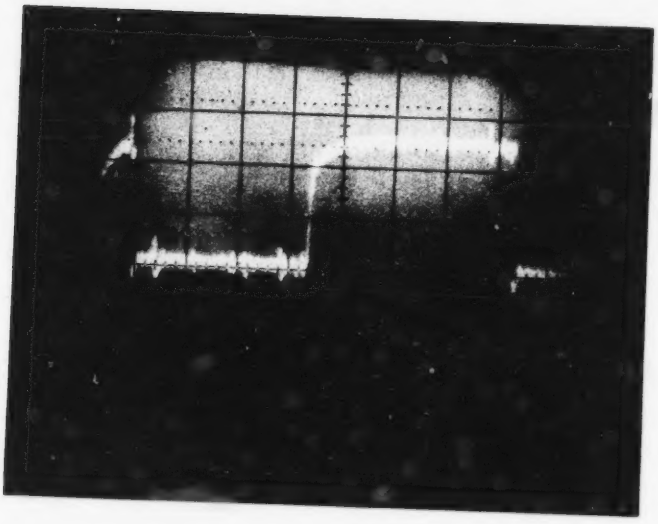
0.2 V/div
time base = 0.1 s

Pin 23 1c24



Bit A4 en opération dyamique
(Phase de programmation du 8237)

Pin 2 1c5

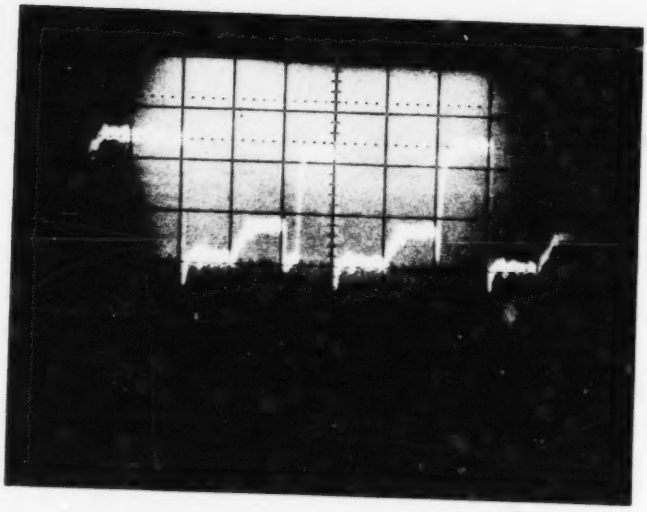


0.2 V/div
time base
0.5 s

D10 Opération Dynamique (Programmation)

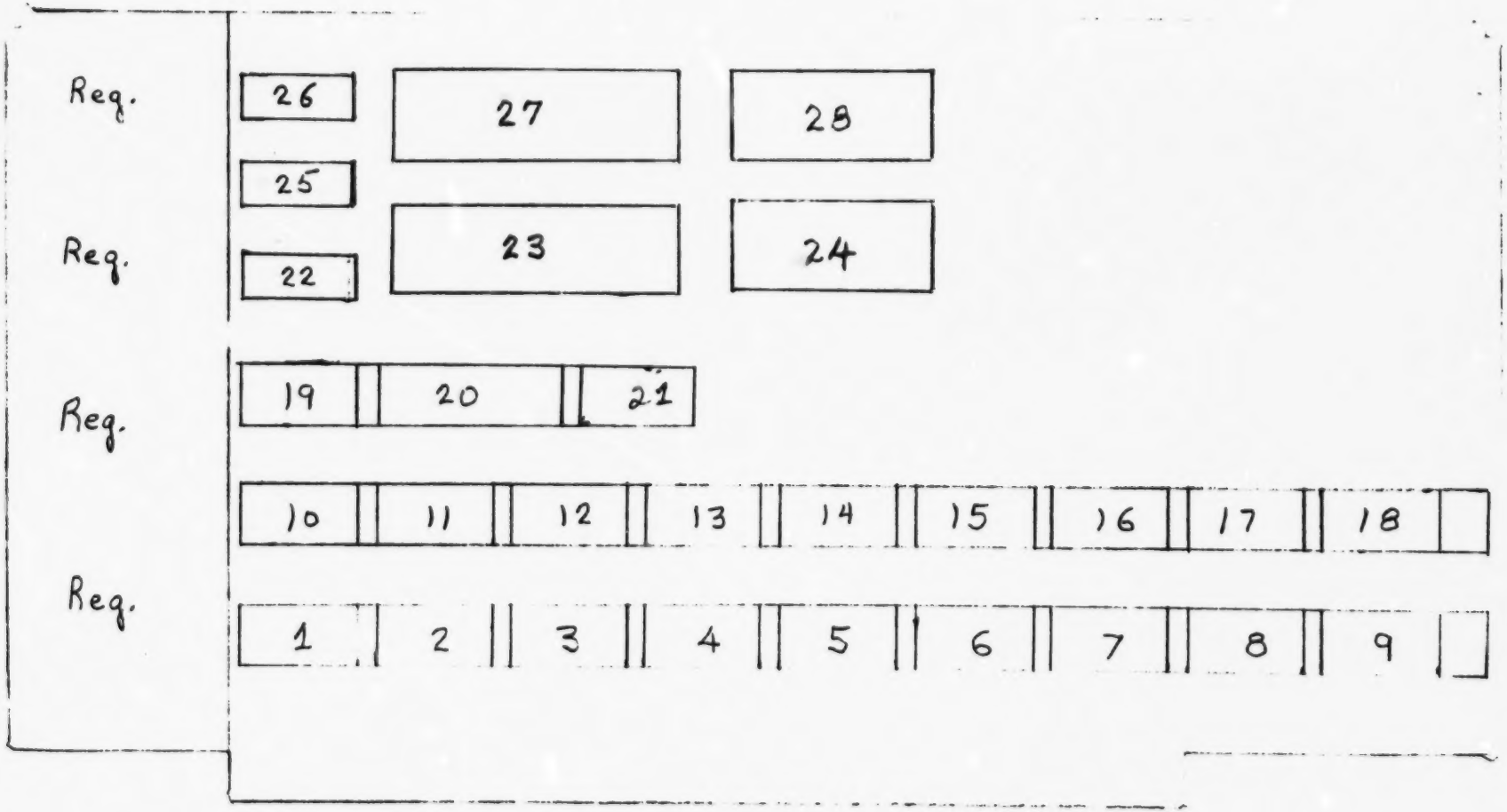
0.2 V/div
time base = 0.5 s

1c5 pin 4

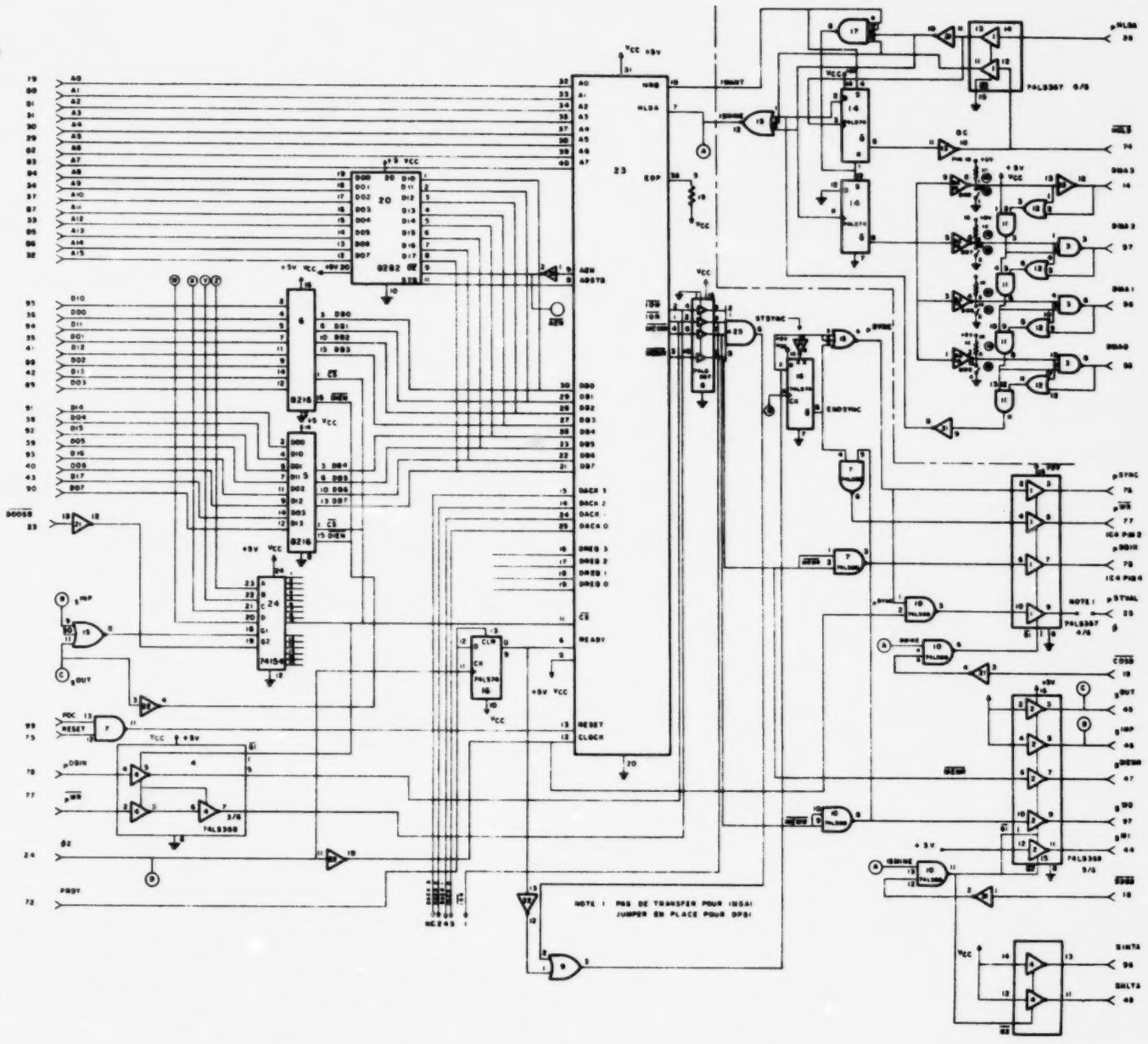


DOO
opération dynamique (Programmation)

ANNEXE E: DISPOSITION DES PIÈCES SUR LE MONTAGE



ANNEXE P: SCHÉMA ÉLECTRONIQUE



POOR

QUALITY

ORIGINAL